RSAC 2025 Conference

Many Voices.
One Community.

SESSION ID: TPV-T09

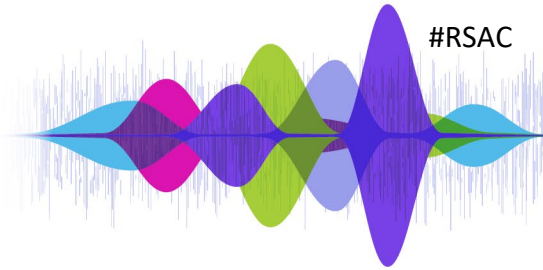
AI, Automation, & Threat Modeling: Lessons Learned from Hacking the Planet

Douglas McKee

Executive Dir. Threat Research, SonicWall
Author and Instructor, SANS
@fulmetalpackets

Ismael Valenzuela

VP Threat Research & Intelligence, Arctic Wolf
Author and Senior Instructor, SANS
@aboutsecurity



Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

About Us

Douglas McKee

16 yrs ‘Hacking All The Things’

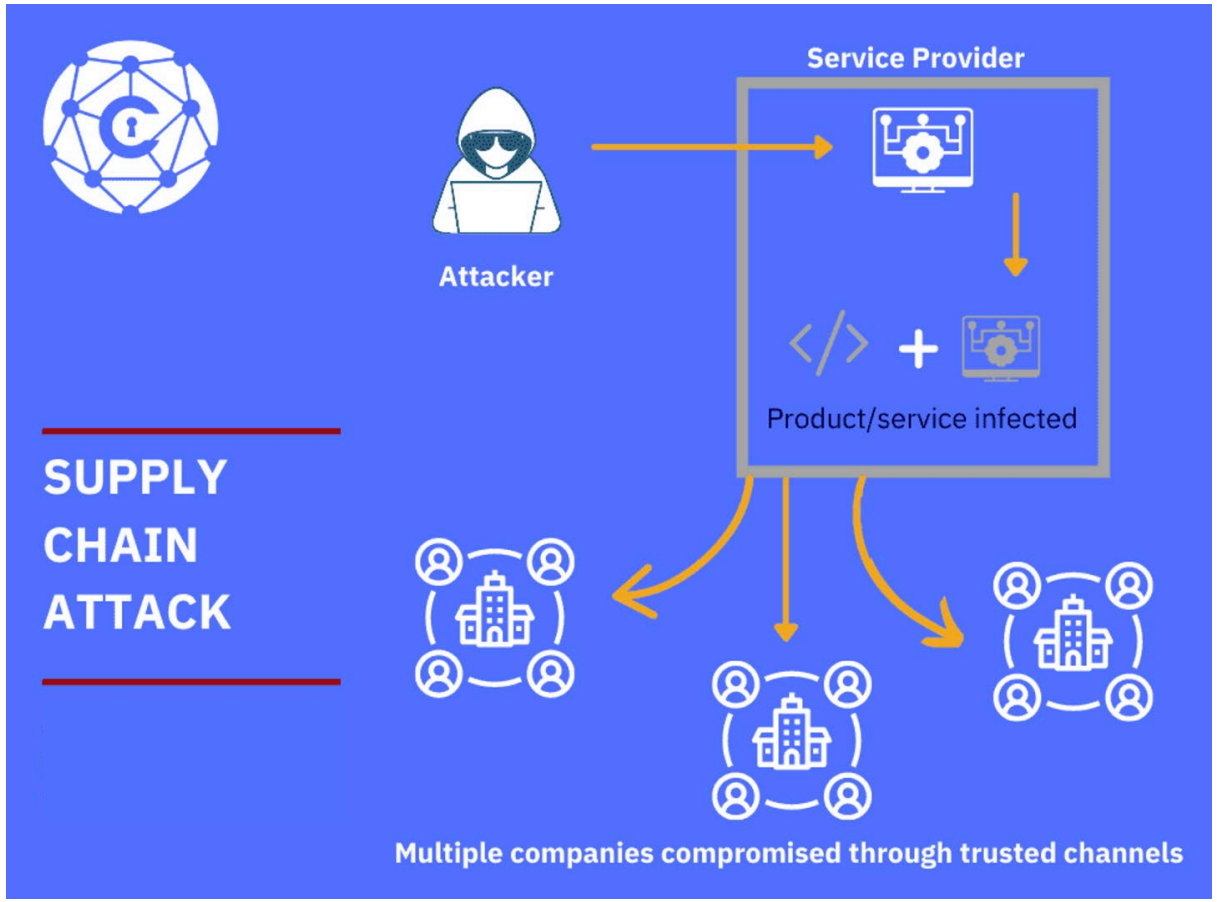
@fulmetalpackets

Ismael Valenzuela

24 yrs. ‘Defending All The Things’

@aboutsecurity

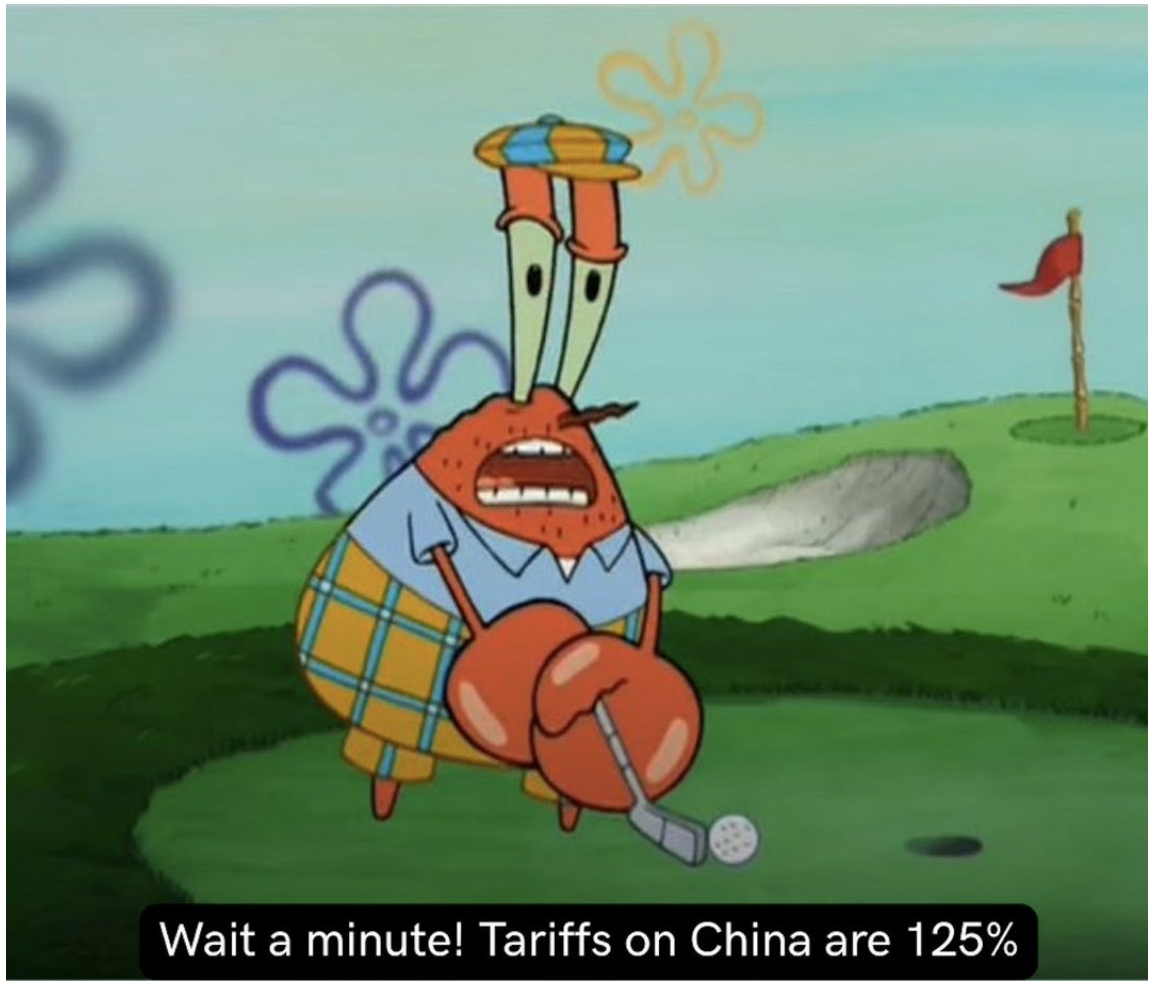
Software Supply Chain Attacks



- Breaches which lead to malicious code modifications
- Delivering malware through legitimate channels
- Known or unknown vulnerabilities in third-party libraries
- Known or unknown vulnerabilities in software development kits

Why Is This Important?

- 'Attackers are lazy'
- And they always look for the highest ROI:
  - Network edge infrastructure (yeah you're doing a 'better' job with your endpoints)
  - and supply chain attacks
- It's not mutually exclusive!
- Current geopolitical and macroeconomic situation may favor this



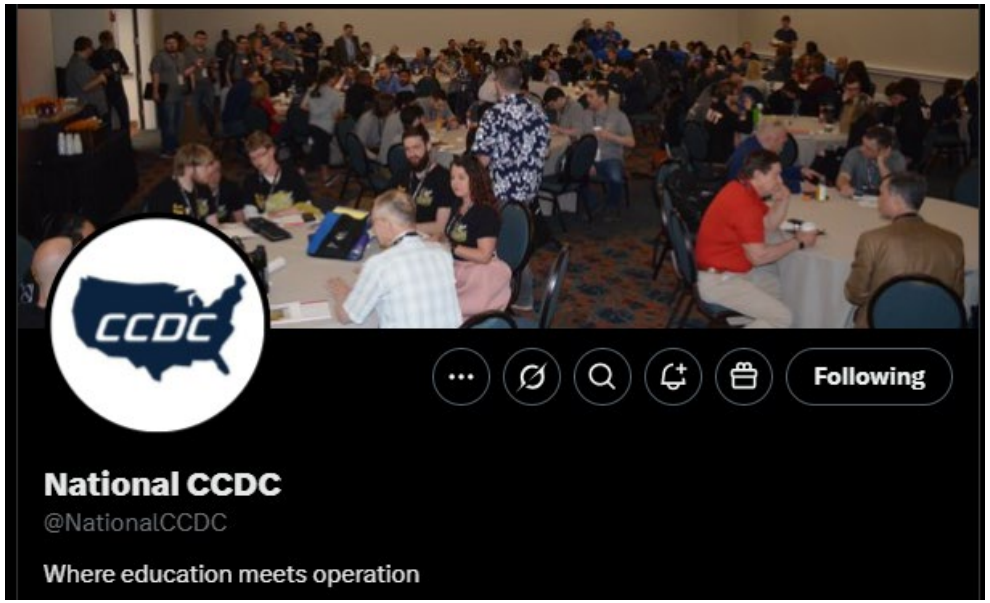
RSAC 2025 Conference

War Stories

Hacking the Planet

Many Voices.
One Community.

Making Students Cry for Fun and Profit



Overview

Pay the Pony marries the idea of ransomware and prankware and creates a new class of software: hilarityware. Pay the Pony was created to inflict hilarity on the students of network security competitions such as the Collegiate Cyber Defense Competitions (CCDC) and the Palmetto Cyber Defense Competition (PCDC). Pay the Pony is comprised of an injector, paythepony.exe, and a DLL, ponyware(32/64).dll.

***https://github.com/whyallyn/paythepony**

From the trenches: weapons evolve, but how about defensive strategy?

Virus paralyses Belgian ministry computers

Published Saturday, February 11, 2012 **By** AFP

A virus has paralysed computers at Belgium's finance ministry and tax offices for two days, Belgian media reported on Saturday.

The virus, which idles or shuts off computers, first appeared on Thursday and has also affected machines at tax and administrative offices throughout the country, reported the Het Belang van Limburg, la Gazet van Antwerpen and De Standaard dailies.

The virus, named "sality.gen" had not managed to breach the data base of individual tax returns at the ministry, the Belga news agency quoted Francis Adyns, spokesman for the federal public finance service, as saying.

Malware Used by China-Backed Hackers

Tuesday, January 14, 2025

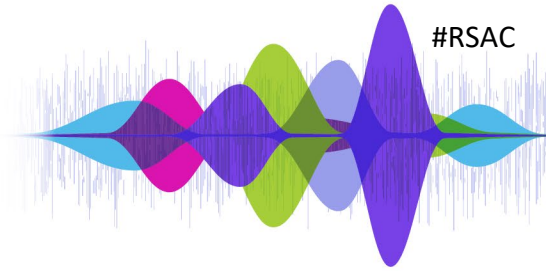
For Immediate Release
Office of Public Affairs

Court-Authorized Operation Removes PlugX Malware from More Than 4,200 Infected U.S. Computers

Note: View the affidavit here.

The Justice Department and FBI today announced a multi-month law enforcement operation that, alongside international partners, deleted “PlugX” malware from thousands of infected computers worldwide. As described in court documents unsealed in the Eastern District of Pennsylvania, a group of hackers sponsored by the People’s Republic of China (PRC), known to the private as “Mustang Panda” and “Twill Typhoon,” used a version of PlugX malware to infect, contro steal information from victim computers.

TOP



Speed and Efficiency Matter

RSAC™ 2025 Conference

Hacking Defensively

Think Red, Act Blue

Many Voices.
One Community.

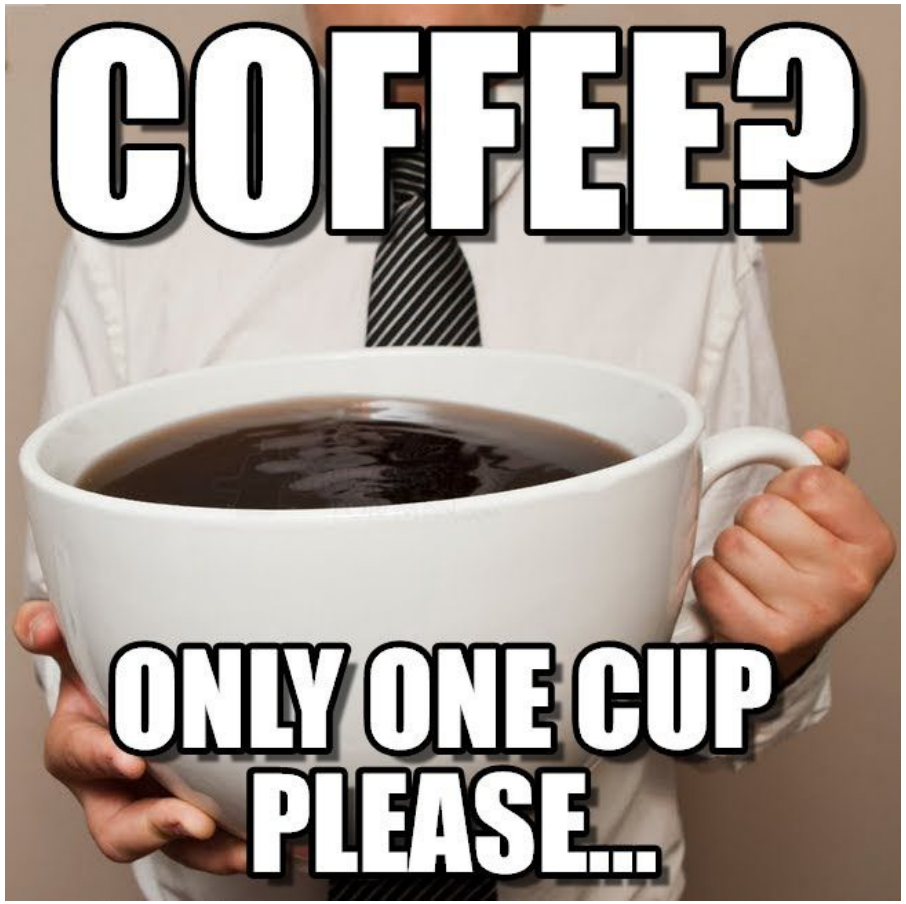
Product Security Testing

- Evaluating a product for new threats introduced to an organization
- Proactive Security measure
  - What risk is introduced?
  - Mitigating measures?
  - **Knowledge** is power

Why Product Security Testing?

Avoid Trial and Error

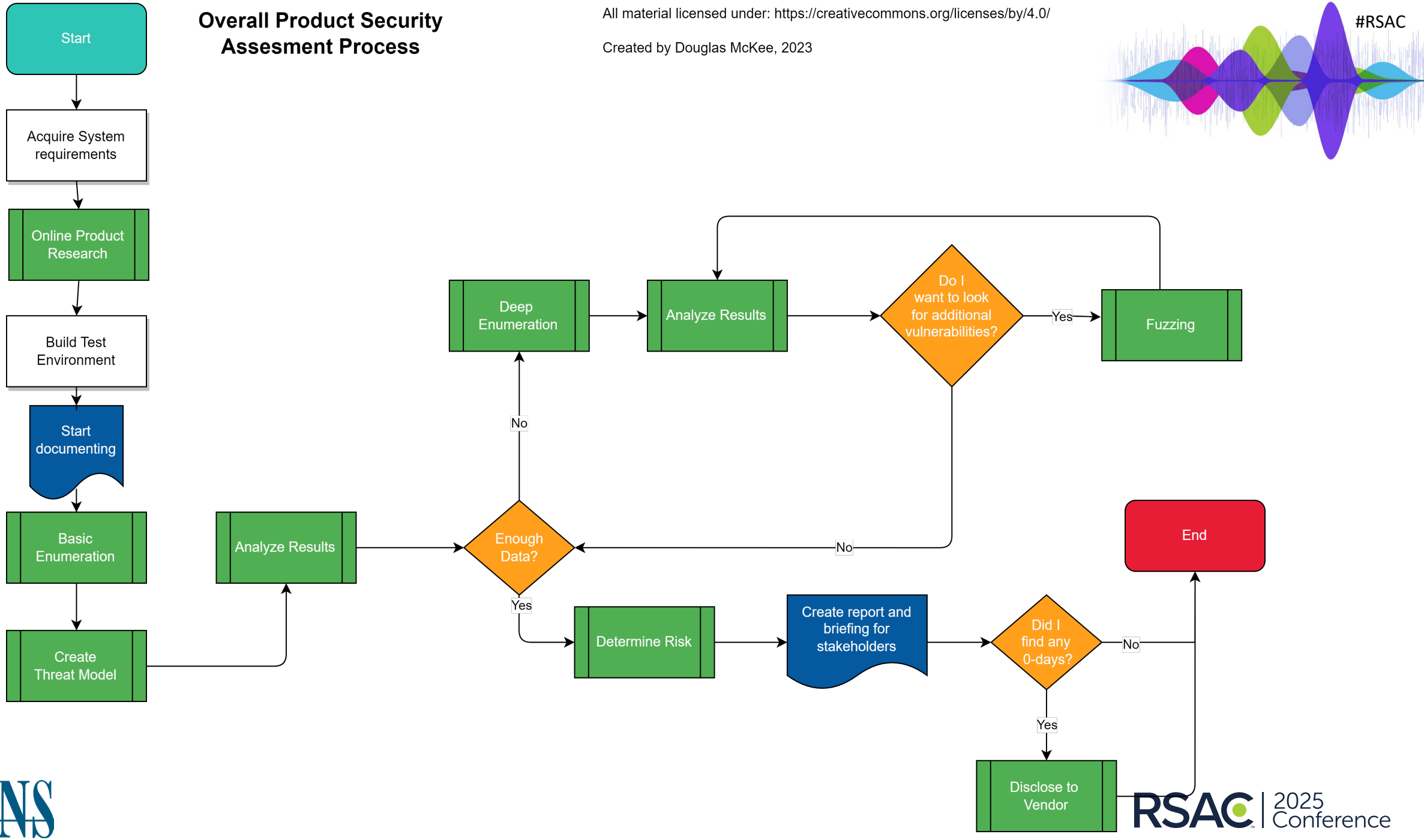
- Is coffee good for you? Answer: it depends on your CYP1A2 genotype
- For fast caffeine metabolizers, coffee is “heart healthy”. Slow coffee metabolizers can have an increased risk of heart attack
- The use of DNA testing can reveal that
- sed ‘s/DNA/Product Security/g'

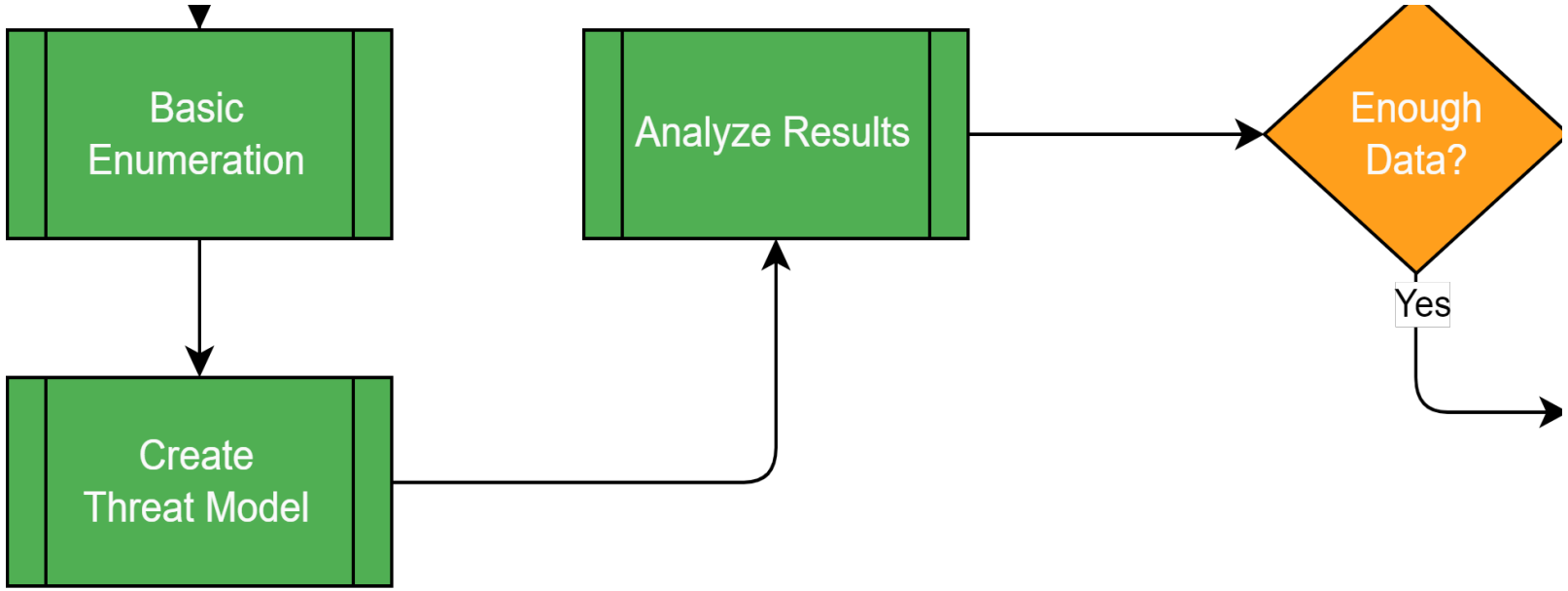


Overall Product Security Assesment Process

All material licensed under: https://creativecommons.org/licenses/by/4.0/

Created by Douglas McKee, 2023




Basic Enumeration
Create Threat Model
Analyze Results
Enough Data?
Yes

RSAC 2025 Conference

Methodology Applied Realistically

AI and Automation Applied to Product Security

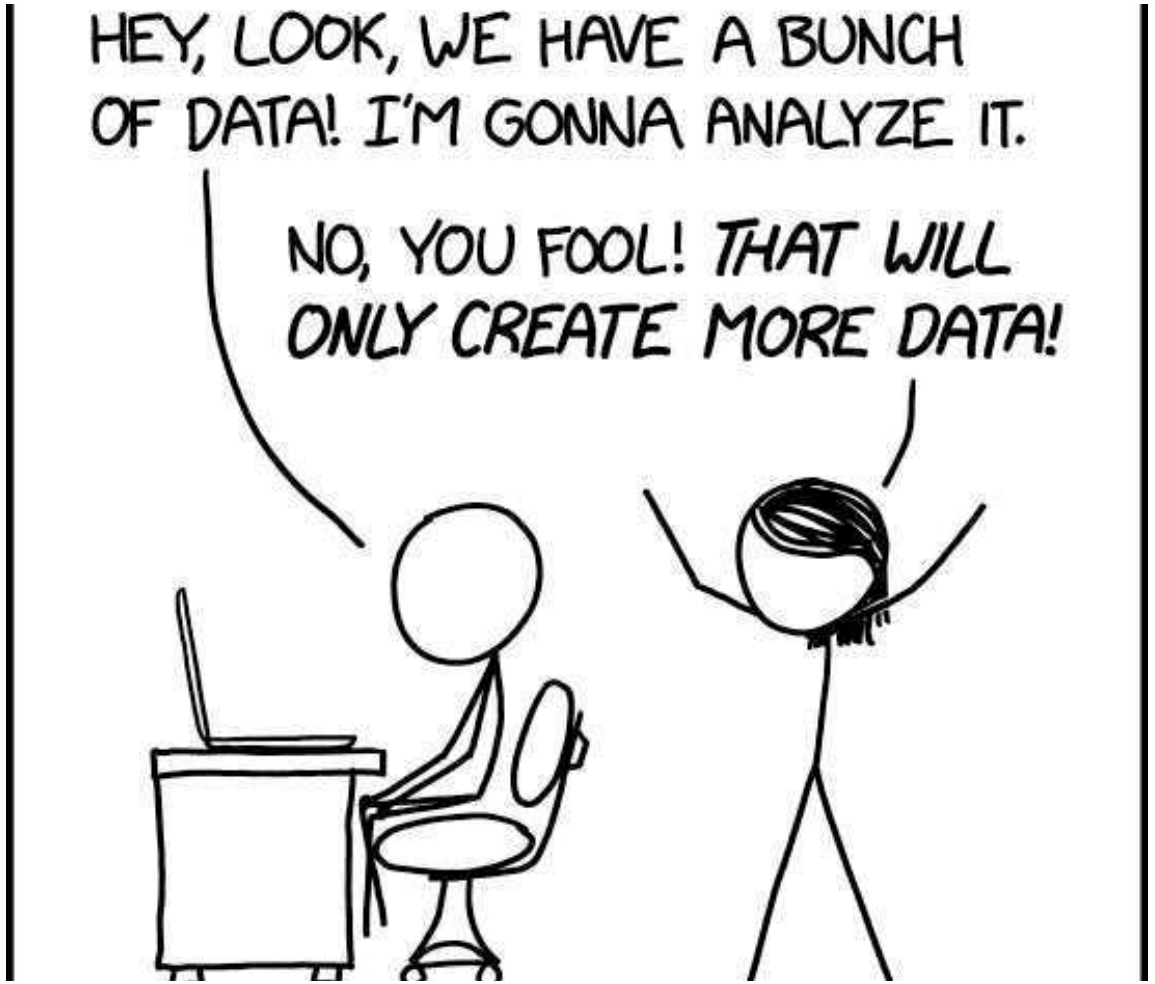
Many Voices.
One Community.

Threat Modeling Manifesto

- Threat modeling analyzes representations of a system to highlight concerns about security and privacy characteristics.
- At the highest levels, when we threat model, we ask four key questions:
    1. What are we working on?
    2. What can go wrong?
    3. What are we going to do about it?
    4. Did we do a good enough job?

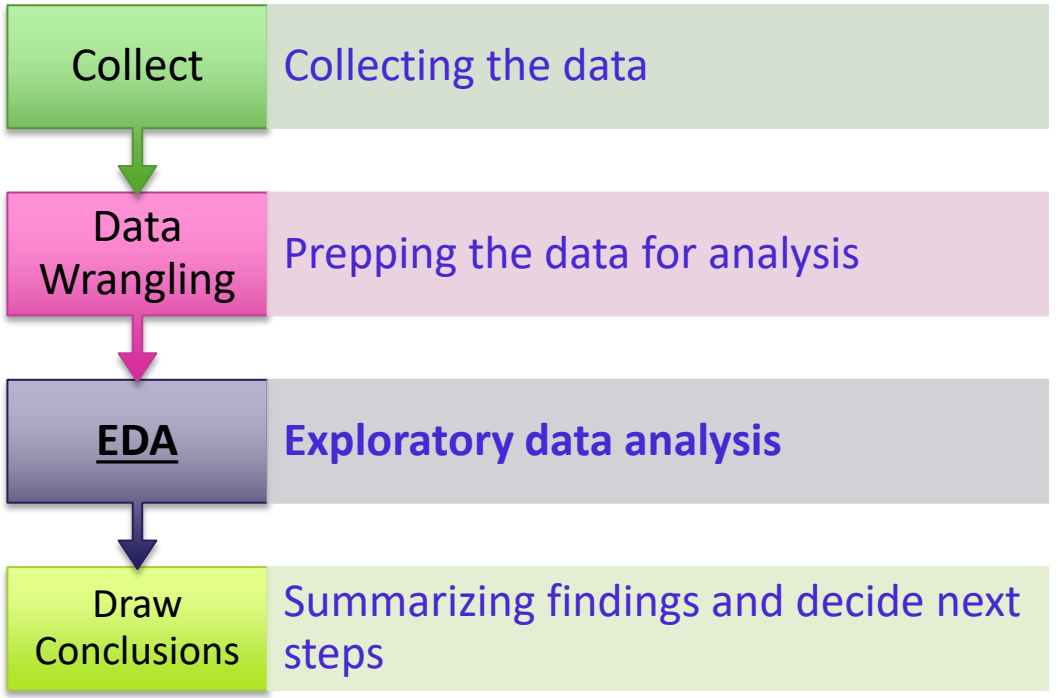
Drowning in Data

1. Gathered TONS of data
2. What Matters?
3. Enrich the data
4. More DATA!
5. Produce a Threat Model

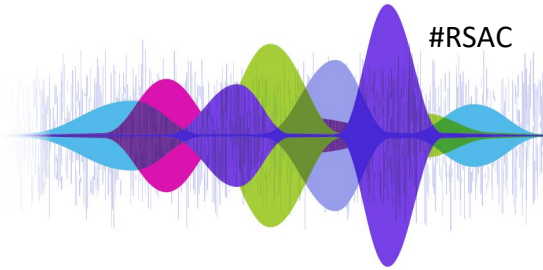


Exploratory Data Analysis (EDA)

- Answering questions related to product security testing requires sifting through large amounts of data.
  - What protocols do “AppX” use for login?
  - Are credentials sent in cleartext?
  - How many servers are contacted, and where are they located?
- These answers are needed to build a **threat model**.
- EDA is a workflow to conduct analysis to make sense of the data and present the results.
- The backbone of this process is **statistics**.



Scenarios



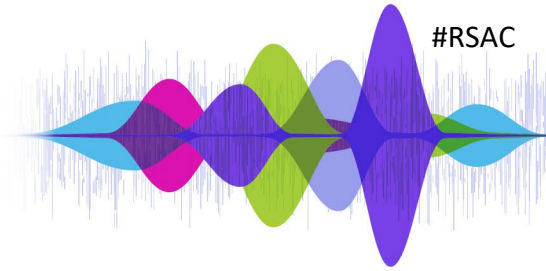
Firmware Basic Enumeration

1. Unpack firmware
2. Statically analyze
3. Enrichment
4. Build a Threat Model

Windows App Basic Enumeration

1. Install Application
2. Dynamically analyze
3. Enrichment
4. Build a Threat Model

Tooling



Leveraging Orchestration

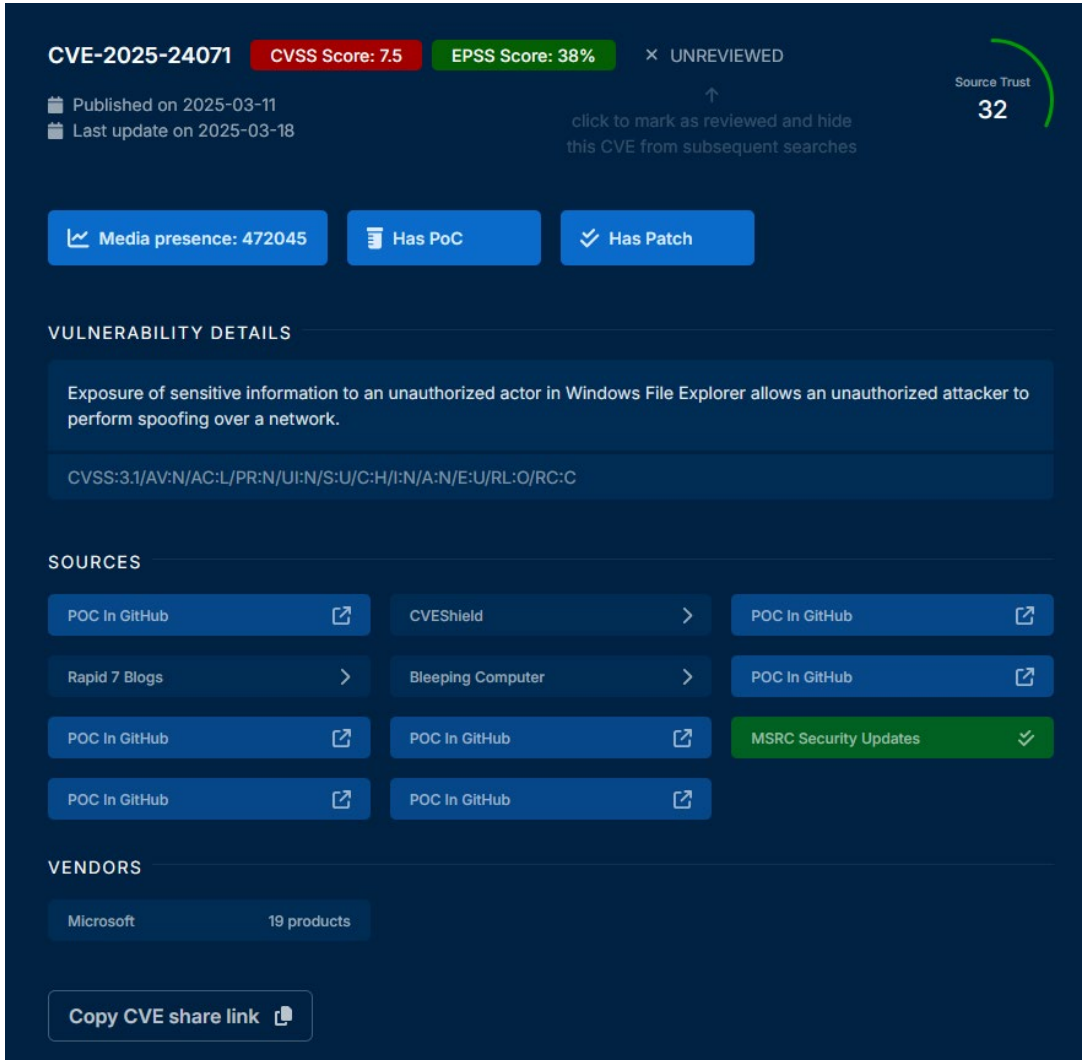
- ReTool
- BinWalk
- Cve-bin-tool
- CVEs details through local database
- LLM Analysis
- **Output**: Threat Model

Leveraging AI Agents

- N8N
- Sysinternal's Procmon
- Qdrant Vector Store
- External CVE enrichment
- LLM Analysis
- **Output**: Threat model

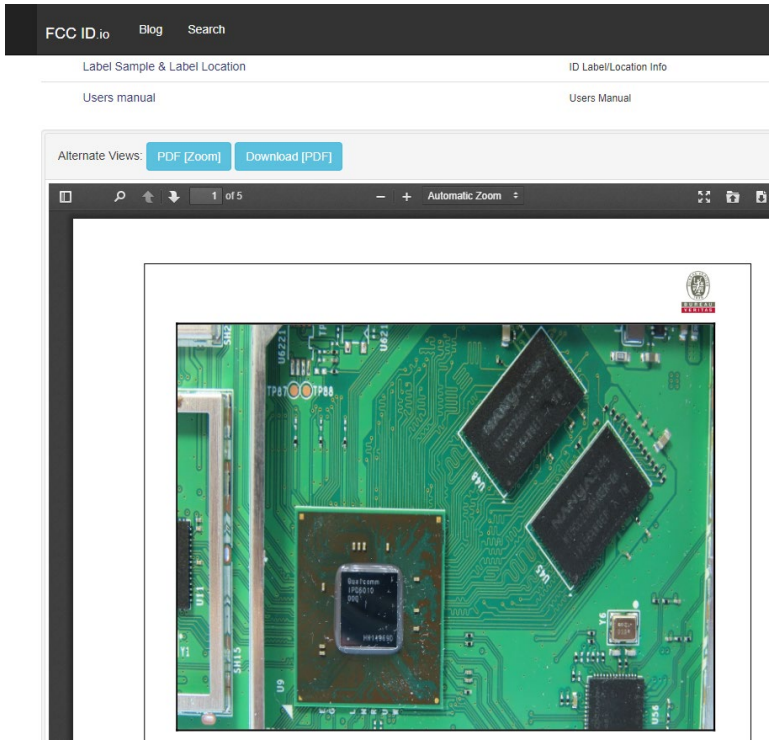
Online Product Research

- Vendor website
- FCC ID documents
- Search engines
- GitHub
- Vulnerability databases
  - nvd.nist.gov
  - cve.mitre.org
  - cvedetails.com
  - vuldb.com
  - exploit-db.com
  - Rapid Risk Radar



Firmware: Online Product Research

Can you process this document from the FCC ID and look at the images and tell me what additional chips and capabilities are included in the UDM Pro? https://fcc.report/FCC-ID/SWX-UDMPRO/4365483

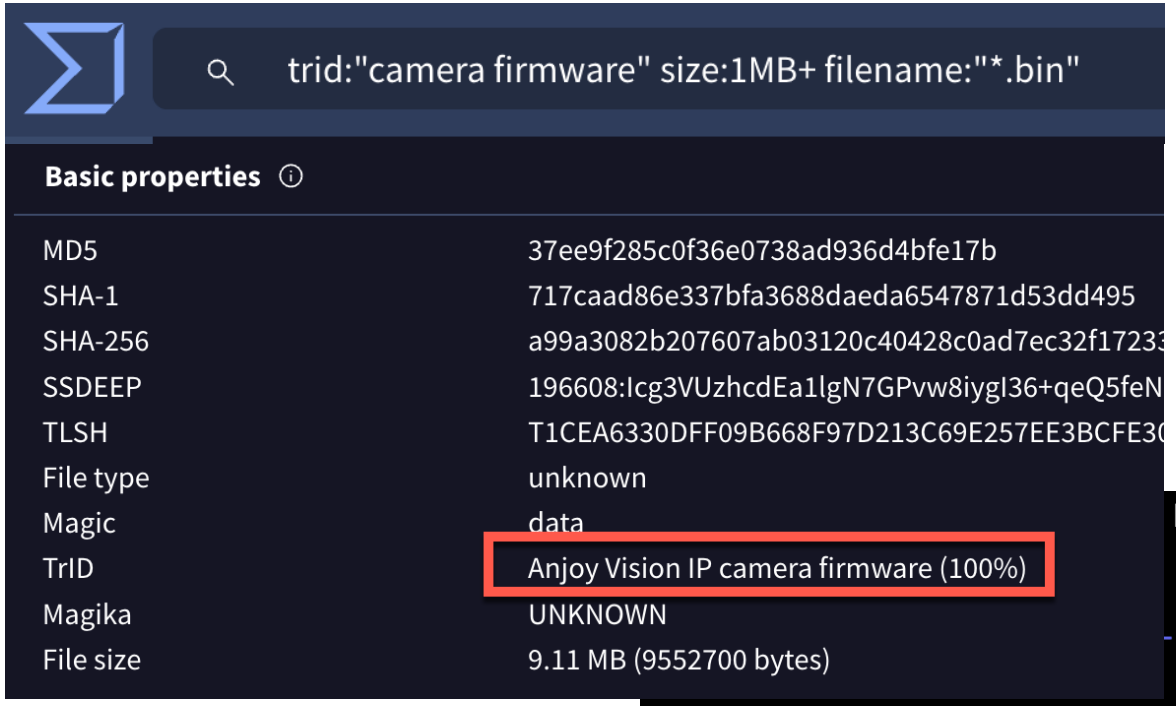


ChatGPT o4-mini-high

Below is a summary of the key "extra" silicon and sub-systems you can see in the UDM Pro's FCC internal-photo package that go beyond the block-diagram in the datasheet:

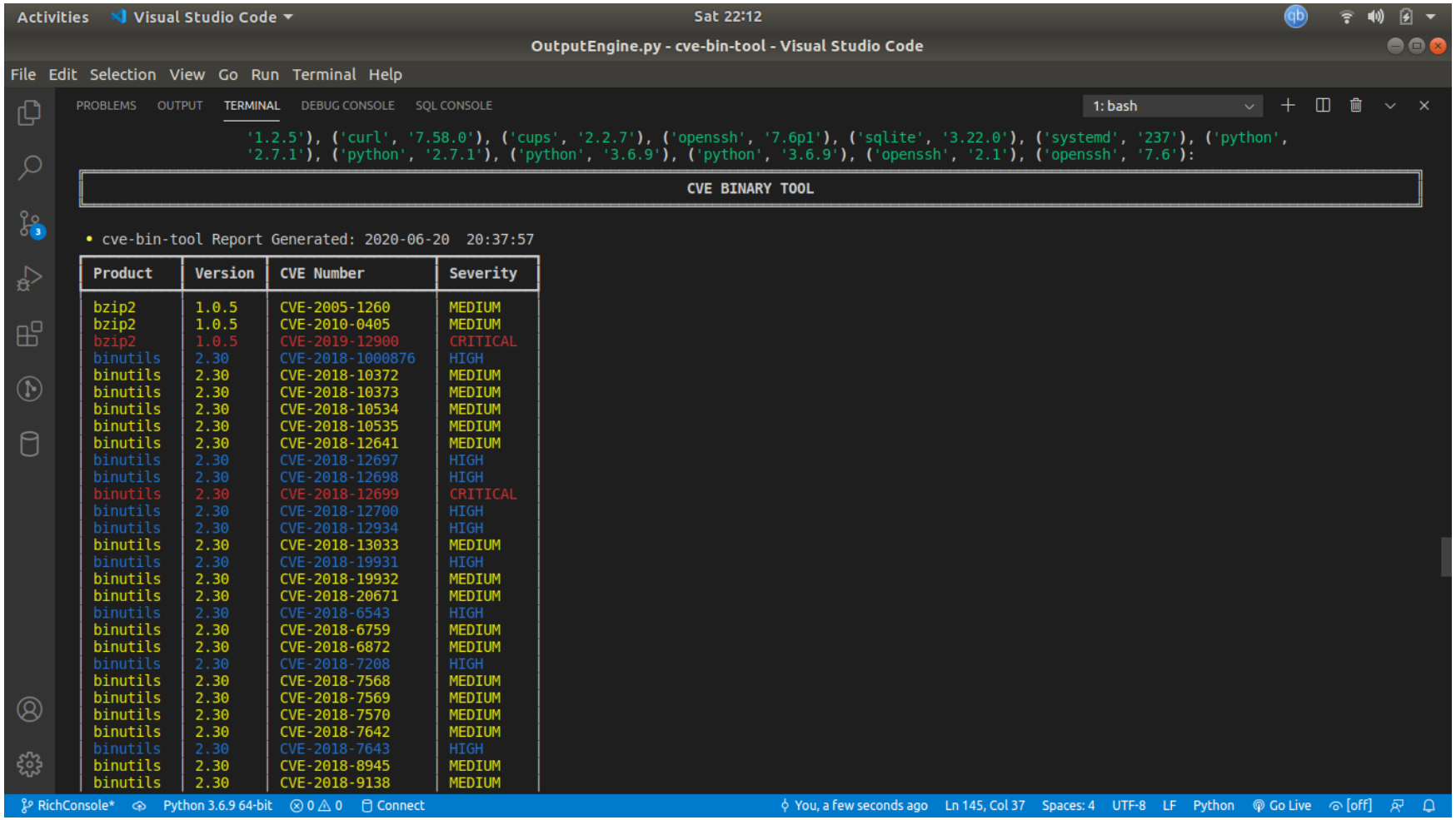
1. **Bluetooth Low-Energy Provisioning Module**
   A discrete BLE radio module (with its own chip antenna and RF filters) sits on a little daughterboard for initial mobile-app adoption over Bluetooth Low Energy
   UniFi Cloud Gateways - Tech Specs .
2. **8-Port Gigabit Switch ASIC**
   An 8×1 GbE switch chip (commonly a Realtek RTL8367S-series device) aggregates the LAN ports and ties them to the SoC over a single 1 Gbps backplane Reddit .
3. **I²C EEPROM for Alpine Bootloader**
   The Cavium/Annapurna Alpine AL-314 SoC uses an I²C-attached EEPROM (at SPD address 0x57) to load its stage-3 boot code. You can spot the little SOIC-8 EEPROM next to the SoC in the photos Rick Mark .
4. **QSPI Serial Flash (U-Boot)**
   A 16 Mb (or similar) serial NOR flash package holds the U-Boot bootloader. In the internal images you can see the characteristic SOIC-8 QSPI flash device close to the CPU Rick Mark .
5. **Power-Management & Regulation**
   Several DC-DC converter modules and a PMIC (likely TI TPS65217 or equivalent) generate the multiple core, I/O, and DDR rails required by the SoC and peripherals Rick Mark .
6. **Fan-Control & Thermal Monitoring**
   A small PWM fan-controller IC (and a companion digital temperature sensor, e.g. TI TMP451) on the I²C bus actively drives the heatsink fan based on SoC and ambient temperatures Rick Mark .

Firmware: Data Gathering




```
binwalk -eM firmware_MC500L8_V0-H5_V3.2.0.6_STARC_202211091037.bin

                                                    'firmwares/extractions/firmware_MC50

DECIMAL                    HEXADECIMAL                DESCRIPTION
--------------------------------------------------------------------------------
1556                       0x614                      uImage firmware image, header size: 6
                                                      load address: 0x20008000, entry point
1906696                    0x1D1808                   SquashFS file system, little endian,
                                                      2022-11-09 02:37:16
--------------------------------------------------------------------------------
[+] Extraction of uimage data at offset 0x614 completed successfully
[+] Extraction of squashfs data at offset 0x1D1808 completed successfully
--------------------------------------------------------------------------------
```

Firmware: Data Enrichment


Activities
Visual Studio Code
Sat 22:12
OutputEngine.py - cve-bin-tool - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE SQL CONSOLE
1: bash
'1.2.5'), ('curl', '7.58.0'), ('cups', '2.2.7'), ('openssh', '7.6p1'), ('sqlite', '3.22.0'), ('systemd', '237'), ('python', '2.7.1'), ('python', '2.7.1'), ('python', '3.6.9'), ('python', '3.6.9'), ('openssh', '2.1'), ('openssh', '7.6'):
CVE BINARY TOOL
• cve-bin-tool Report Generated: 2020-06-20 20:37:57
Product | Version | CVE Number | Severity
bzip2 | 1.0.5 | CVE-2005-1260 | MEDIUM
bzip2 | 1.0.5 | CVE-2010-0405 | MEDIUM
bzip2 | 1.0.5 | CVE-2019-12900 | CRITICAL
binutils | 2.30 | CVE-2018-1000876 | HIGH
binutils | 2.30 | CVE-2018-10372 | MEDIUM
binutils | 2.30 | CVE-2018-10373 | MEDIUM
binutils | 2.30 | CVE-2018-10534 | MEDIUM
binutils | 2.30 | CVE-2018-10535 | MEDIUM
binutils | 2.30 | CVE-2018-12641 | MEDIUM
binutils | 2.30 | CVE-2018-12697 | HIGH
binutils | 2.30 | CVE-2018-12698 | HIGH
binutils | 2.30 | CVE-2018-12699 | CRITICAL
binutils | 2.30 | CVE-2018-12700 | HIGH
binutils | 2.30 | CVE-2018-12934 | HIGH
binutils | 2.30 | CVE-2018-13033 | MEDIUM
binutils | 2.30 | CVE-2018-19931 | HIGH
binutils | 2.30 | CVE-2018-19932 | MEDIUM
binutils | 2.30 | CVE-2018-20671 | MEDIUM
binutils | 2.30 | CVE-2018-6543 | HIGH
binutils | 2.30 | CVE-2018-6759 | MEDIUM
binutils | 2.30 | CVE-2018-6872 | MEDIUM
binutils | 2.30 | CVE-2018-7208 | HIGH
binutils | 2.30 | CVE-2018-7568 | MEDIUM
binutils | 2.30 | CVE-2018-7569 | MEDIUM
binutils | 2.30 | CVE-2018-7570 | MEDIUM
binutils | 2.30 | CVE-2018-7642 | MEDIUM
binutils | 2.30 | CVE-2018-7643 | HIGH
binutils | 2.30 | CVE-2018-8945 | MEDIUM
binutils | 2.30 | CVE-2018-9138 | MEDIUM
RichConsole* Python 3.6.9 64-bit 0 0 Connect
You, a few seconds ago Ln 145, Col 37 Spaces: 4 UTF-8 LF Python Go Live [off]

Firmware: Automation

```
(.venv) ➜  firmnightware git:(main) ✗ streamlit run firmnightware.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.1.42:8501
```

Vulnerability Report Dashboard

Upload a JSON file to display the dashboard.

Upload Vulnerability JSON File

Drag and drop file here

Limit 200MB per file • JSON

Browse files

sample-report.json 1.3MB

Vulnerability Report Dashboard

Tool: cve-bin-tool v3.4

Generated on: 2025-02-24 03:35:35

Database Last Updated: 2025-02-24 03:30:16

Total Entries: - **NVD:** 282133

- **GAD:** 23802
- **REDHAT:** 19392

Vulnerabilities Summary

	Severity	Count
0	CRITICAL	15
1	HIGH	641
2	MEDIUM	1436
3	LOW	42
4	UNKNOWN	0
5	CRITICAL-EXPLOIT	0
6	HIGH-EXPLOIT	4
7	MEDIUM-EXPLOIT	0
8	LOW-EXPLOIT	0

Firmware: Basic LLM Integration

firmnightware

Latest version deployed | Run | Deploy

startTrigger

Enabled triggers

Webhook

Edit triggers

Test JSON Parameters

```
{"cve":"CVE-2024-53104"}
```

AIQuery

Retool AI | Edit

Action: Generate text

Examples: Product description

Input *

```
Context: {{startTrigger.data.cve}}
```

System Message

```
You are an advanced vulnerability researcher. I will
provide you a cve and additional context you will
create me an appropriate threat model I with an
attack tree with most plausible scenarios

You will structure your response in the following
format and output new lines appropriately. Be precise
and provide the best investigation guidance but
whilst being straight to the point with your advise.

### Attack Tree
Most Plausible Scenarios

# Analysis

DREAD Threat Model


### Investigation of {{startTrigger.data.cve}}
explain what is {{startTrigger.data.cve}}
```

Temperature: 0.1

Use Retool Vectors to provide more context to your query

Select Vector

Filters: No vectors selected

Model: gpt-4o-mini

Inputs Data JSON Settings

code1

JavaScript | Python

```
const normalizeAIQueryData = () => {
    const input = AIQuery.data;
    // Normalize the input data here
    return input;
};

return normalizeAIQueryData();
```

Inputs Data JSON Settings

webhookReturn1

Status code: 200

Return body

```
code1.data
```

Inputs Data JSON Settings

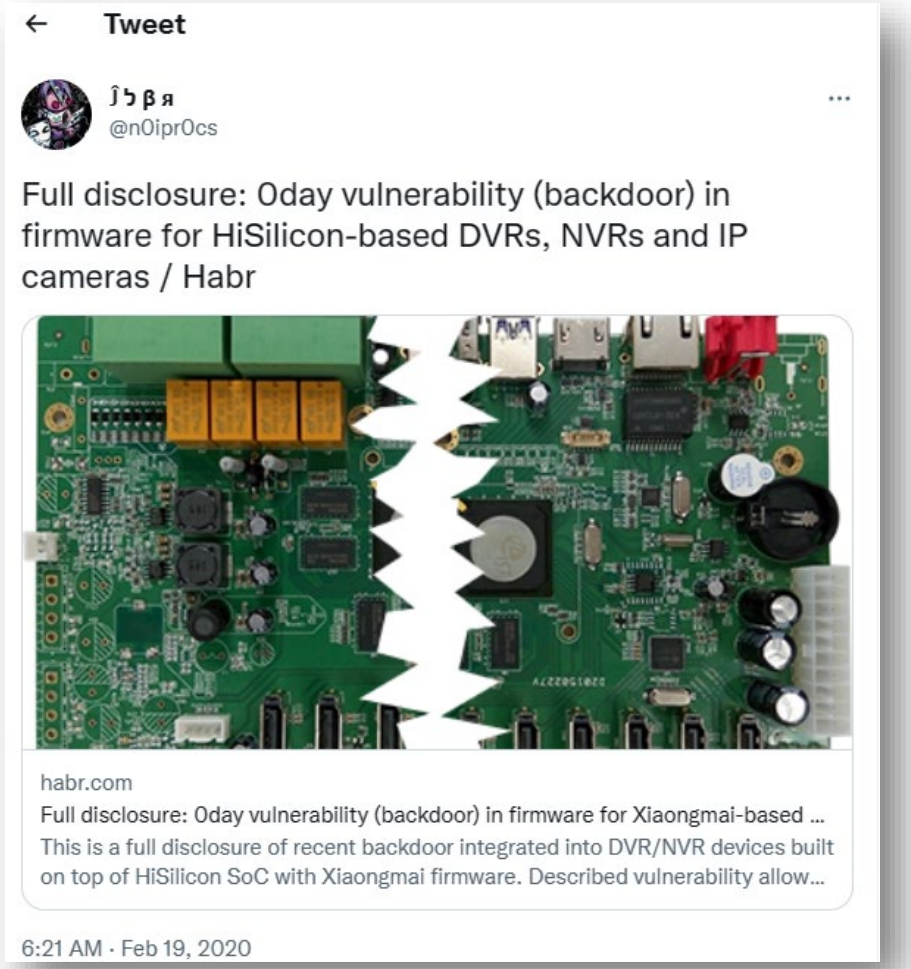
69%

Wait! Is that all?

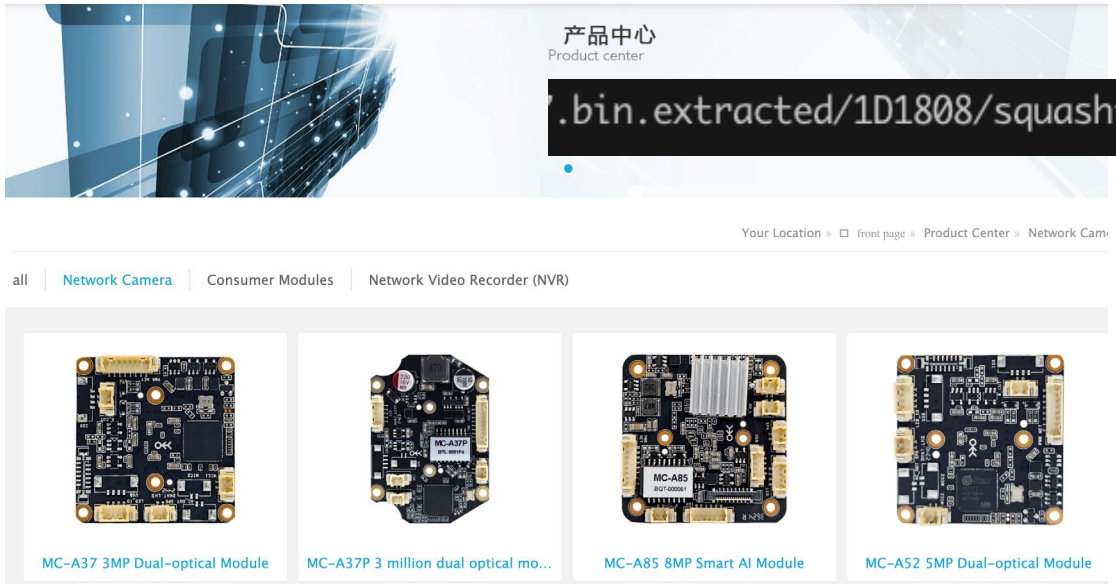
- Let's bring back the 'expert in the loop'
- We have:
  - A Linux based firmware image
  - From a Chinese ODM of embedded video solutions
  - Intended to be used in **IP enabled** cameras
- What are we missing?

‘Backdoors’ in Embedded Devices

- HiSilicon is a “system on a chip” (or SoC) manufacturer, and some of its products are intended for use in IP-enabled video equipment.
- A backdoor was uncovered in 2020 by Vladislav Yarmak.
- His report explains how the devices will activate a telnet server when they receive a “secret knock” sent on port 9530/tcp.


Tweet
@n0ipr0cs
Full disclosure: 0day vulnerability (backdoor) in firmware for HiSilicon-based DVRs, NVRs and IP cameras / Habr
habr.com
Full disclosure: 0day vulnerability (backdoor) in firmware for Xiaongmai-based ...
This is a full disclosure of recent backdoor integrated into DVR/NVR devices built on top of HiSilicon SoC with Xiaongmai firmware. Described vulnerability allow...
6:21 AM · Feb 19, 2020

Grepping All the Things




```
.bin.extracted/1D1808/squashfs-root/opt/ch# strings mainctrl | grep 202.96. -A 5 -B 5
```

```
ifconfig eth0 hw ether %s
route add default gw %s
202.96.134.133
202.96.128.86
touch /tmp/flag.network_configed_ok
```

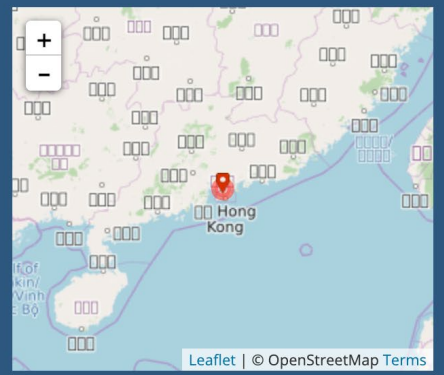
```
ARC_202211091037.bin.extracted/1D1808/squashfs-root/opt/ch# grep -RIn '202\.
96\.134\.133' .
./config.default.mc500l8.xml:30:        <LANConfig ALLNET="1" MacAddress="00
:00:23:34:45:66" DHCP="0" IPAddress="192.168.0.123" Netmask="255.255.255.0"
Gateway="192.168.0.1" DNS1="202.96.134.133" DNS2="8.8.8.8" />
```

SANS

Where Is This Traffic Going To?

IP Details For: 202.96.134.133

Decimal:	3395323525
Hostname:	ns.szptt.net.cn
ASN:	4134
ISP:	Shenzhen
Services:	Datacenter
Country:	China
State/Region:	Guangdong
City:	Shenzhen
Latitude:	22.5457 (22° 32′ 44.42″ N)
Longitude:	114.0681 (114° 4′ 5.19″ E)

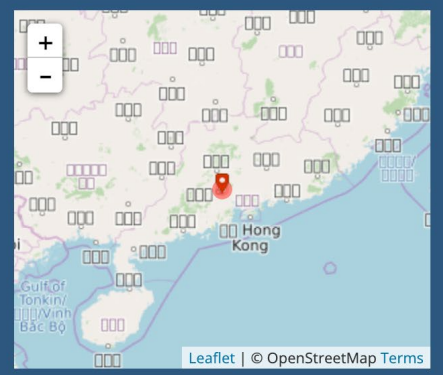


CLICK TO CHECK BLACKLIST STATUS

Latitude and Longitude are often near the center of population. These values are not precise enough to be used to identify a specific address, individual, or for legal purposes. IP data from IP2Location.

IP Details For: 202.96.128.86

Decimal:	3395321942
Hostname:	cache-a.guangzhou.gd.cn
ASN:	4134
ISP:	ChinaNet Guangdong Province Network
Services:	Suspected proxy server
Country:	China
State/Region:	Guangdong
City:	Guangzhou
Latitude:	23.1274 (23° 7′ 38.50″ N)
Longitude:	113.2646 (113° 15′ 52.46″ E)

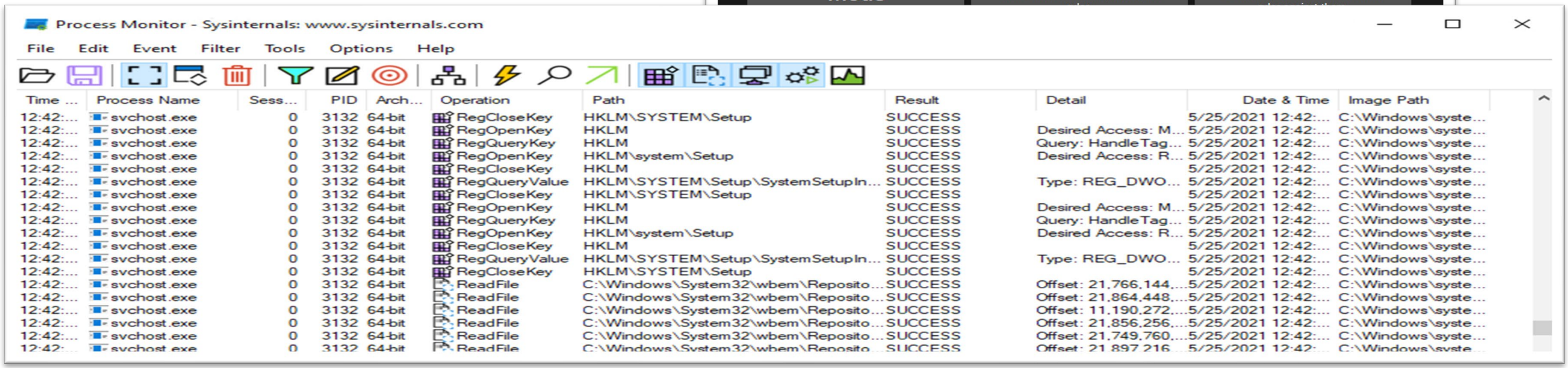


CLICK TO CHECK BLACKLIST STATUS

Latitude and Longitude are often near the center of population. These values are not precise enough to be used to identify a specific address, individual, or for legal purposes. IP data from IP2Location.

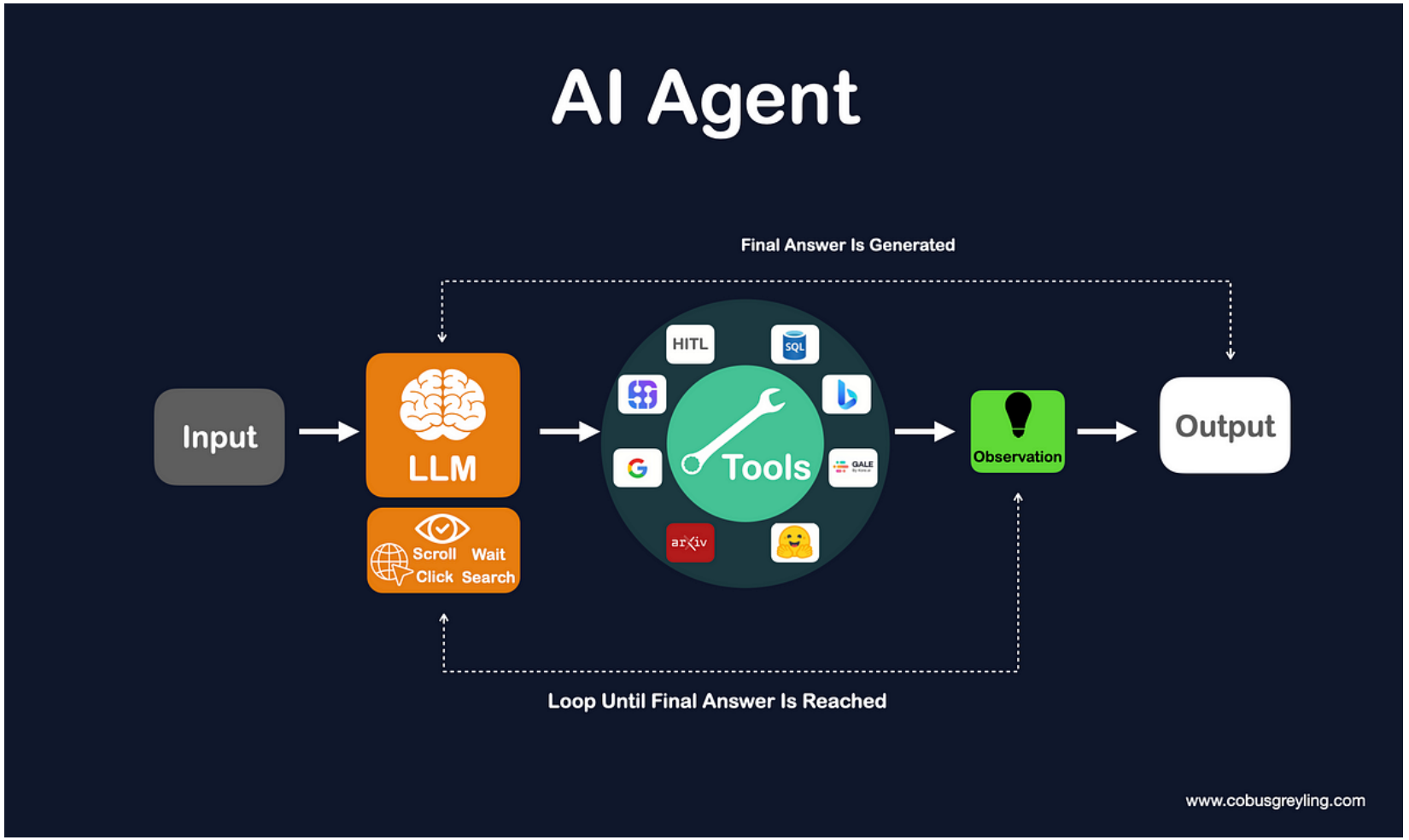
Windows: Data Gathering

- Dynamic Analysis
- ProcMon
- ASA

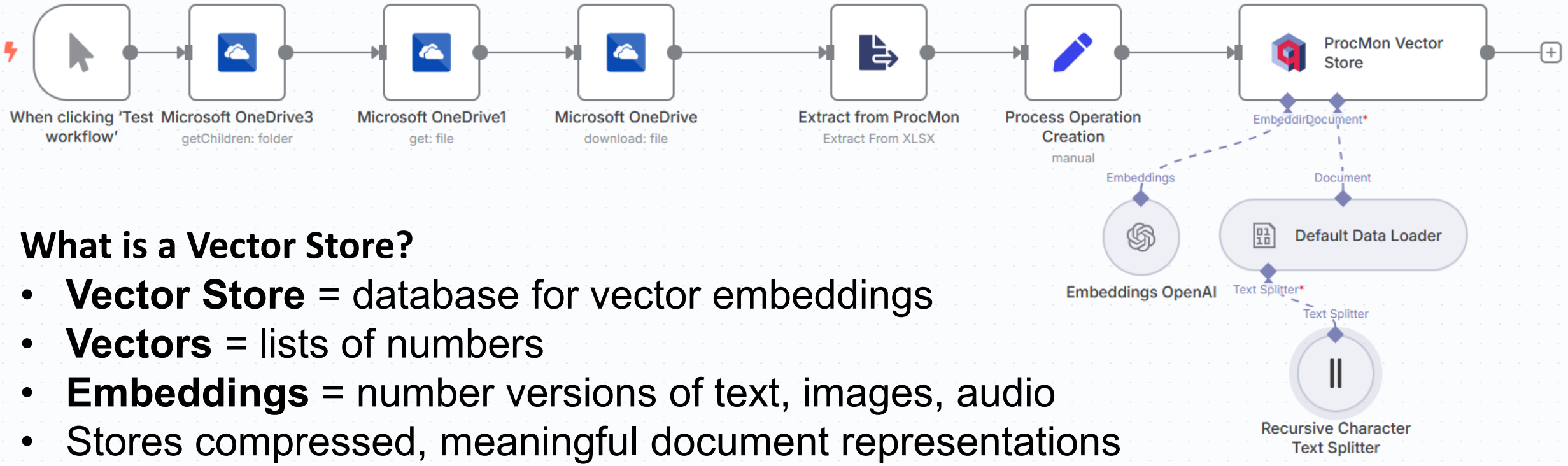

Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help
Time ... Process Name Sess... PID Arch... Operation Path Result Detail Date & Time Image Path
12:42:... svchost.exe 0 3132 64-bit RegCloseKey HKLM\SYSTEM\Setup SUCCESS 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit RegOpenKey HKLM SUCCESS Desired Access: M... 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit RegQueryKey HKLM SUCCESS Query: HandleTag... 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit RegOpenKey HKLM\system\Setup SUCCESS Desired Access: R... 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit RegCloseKey HKLM SUCCESS 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit RegQueryValue HKLM\SYSTEM\Setup\SystemSetupIn... SUCCESS Type: REG_DWO... 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit RegCloseKey HKLM\SYSTEM\Setup SUCCESS 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit RegOpenKey HKLM SUCCESS Desired Access: M... 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit RegQueryKey HKLM SUCCESS Query: HandleTag... 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit RegOpenKey HKLM\system\Setup SUCCESS Desired Access: R... 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit RegCloseKey HKLM SUCCESS 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit RegQueryValue HKLM\SYSTEM\Setup\SystemSetupIn... SUCCESS Type: REG_DWO... 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit RegCloseKey HKLM\SYSTEM\Setup SUCCESS 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit ReadFile C:\Windows\System32\wbem\Reposito... SUCCESS Offset: 21,766,144,... 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit ReadFile C:\Windows\System32\wbem\Reposito... SUCCESS Offset: 21,864,448,... 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit ReadFile C:\Windows\System32\wbem\Reposito... SUCCESS Offset: 11,190,272,... 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit ReadFile C:\Windows\System32\wbem\Reposito... SUCCESS Offset: 21,856,256,... 5/25/2021 12:42:... C:\Windows\syste...
12:42:... svchost.exe 0 3132 64-bit ReadFile C:\Windows\System32\wbem\Reposito... SUCCESS Offset: 21,749,760,... 5/25/2021 12:42:... C:\Windows\syste...

AI Assisted Basic Enumeration: Windows

- What is an AI Agent
  - Perceive its environment
  - Make decisions
  - Take actions
- Tools
  - Rapid Risk Radar
  - Qdrant Vector Store



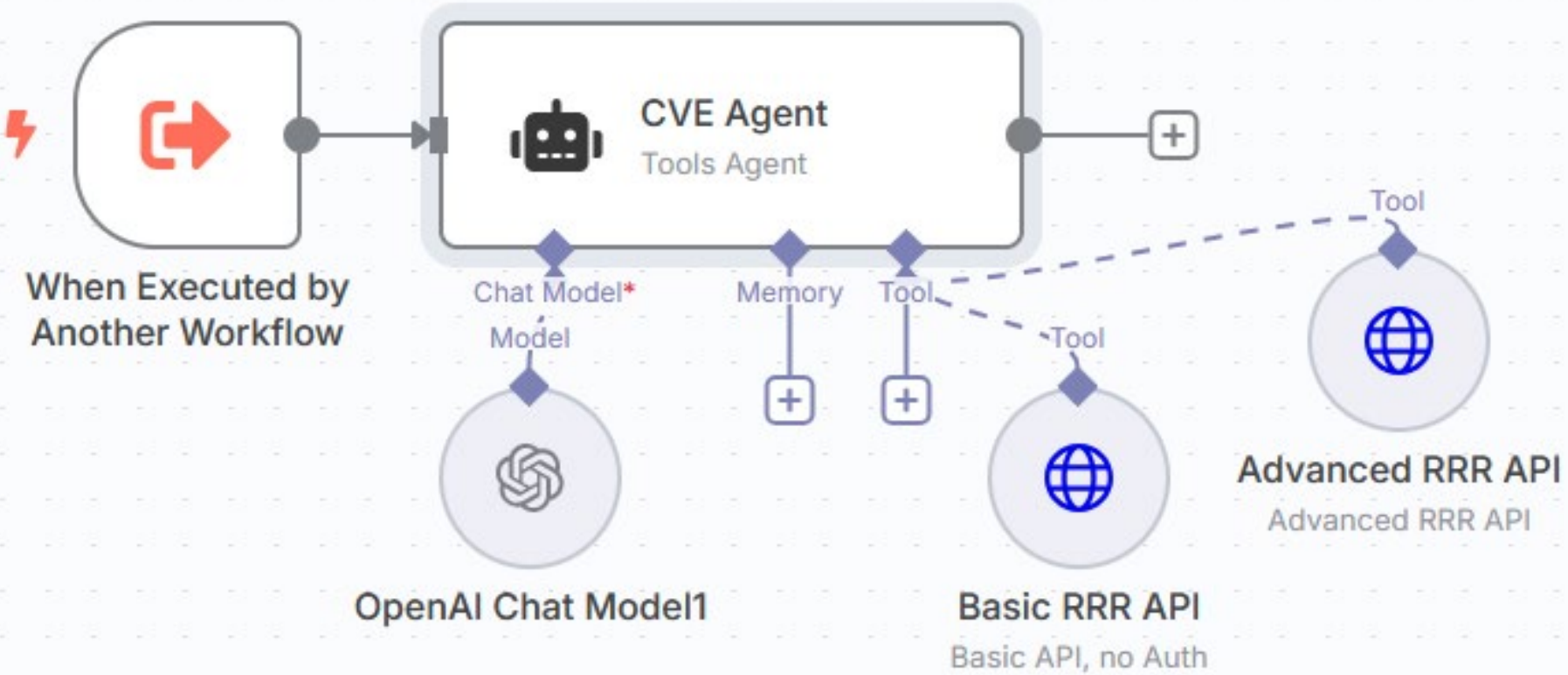
Windows: Data Wrangling



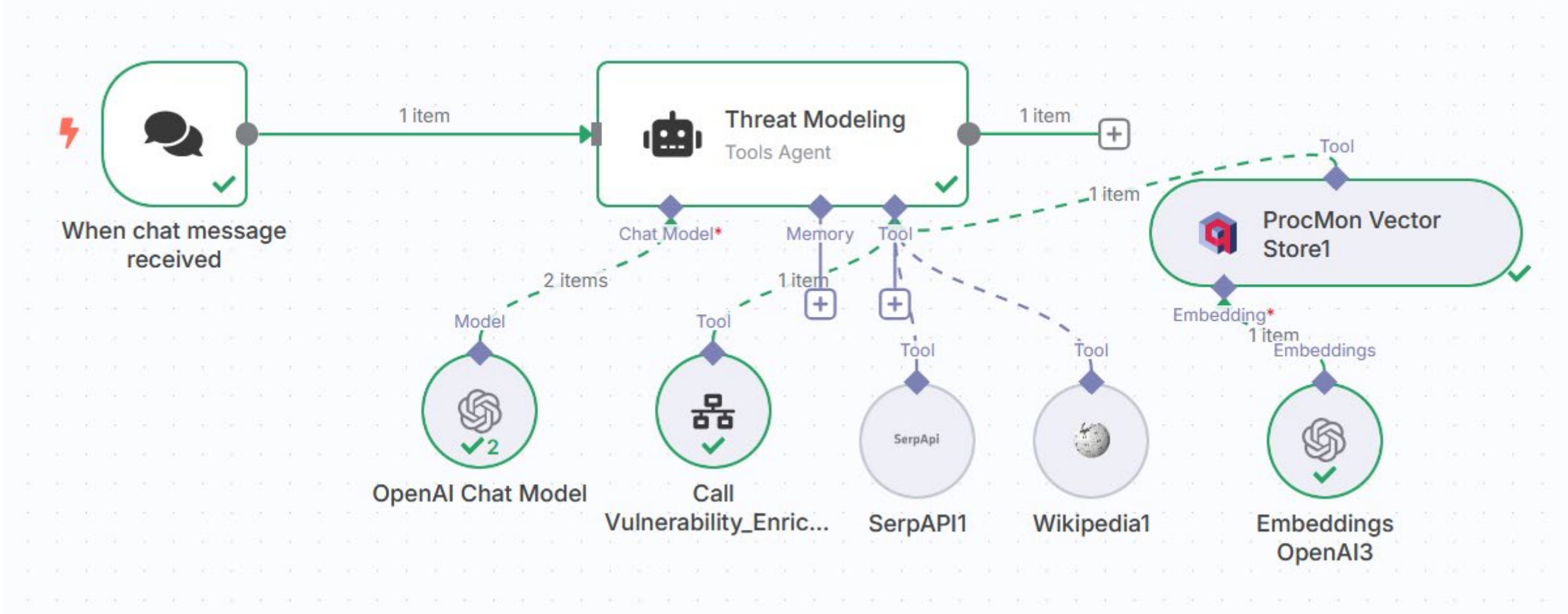
What is a Vector Store?

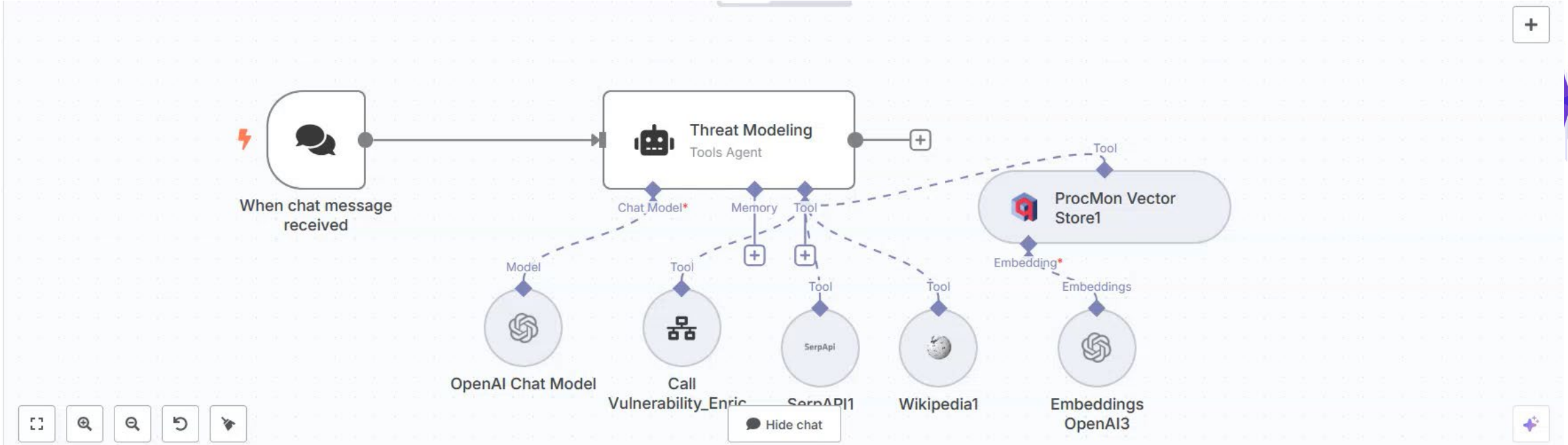
- **Vector Store** = database for vector embeddings
- **Vectors** = lists of numbers
- **Embeddings** = number versions of text, images, audio
- Stores compressed, meaningful document representations
- AI searches by measuring "closeness" (cosine, Euclidean)

Windows: Data Enrichment



Windows: AI Threat Modeling





Chat

Session fcbae7903b214eb4b9d5dc765d2eb8c7

Latest Logs from Threat Modeling node

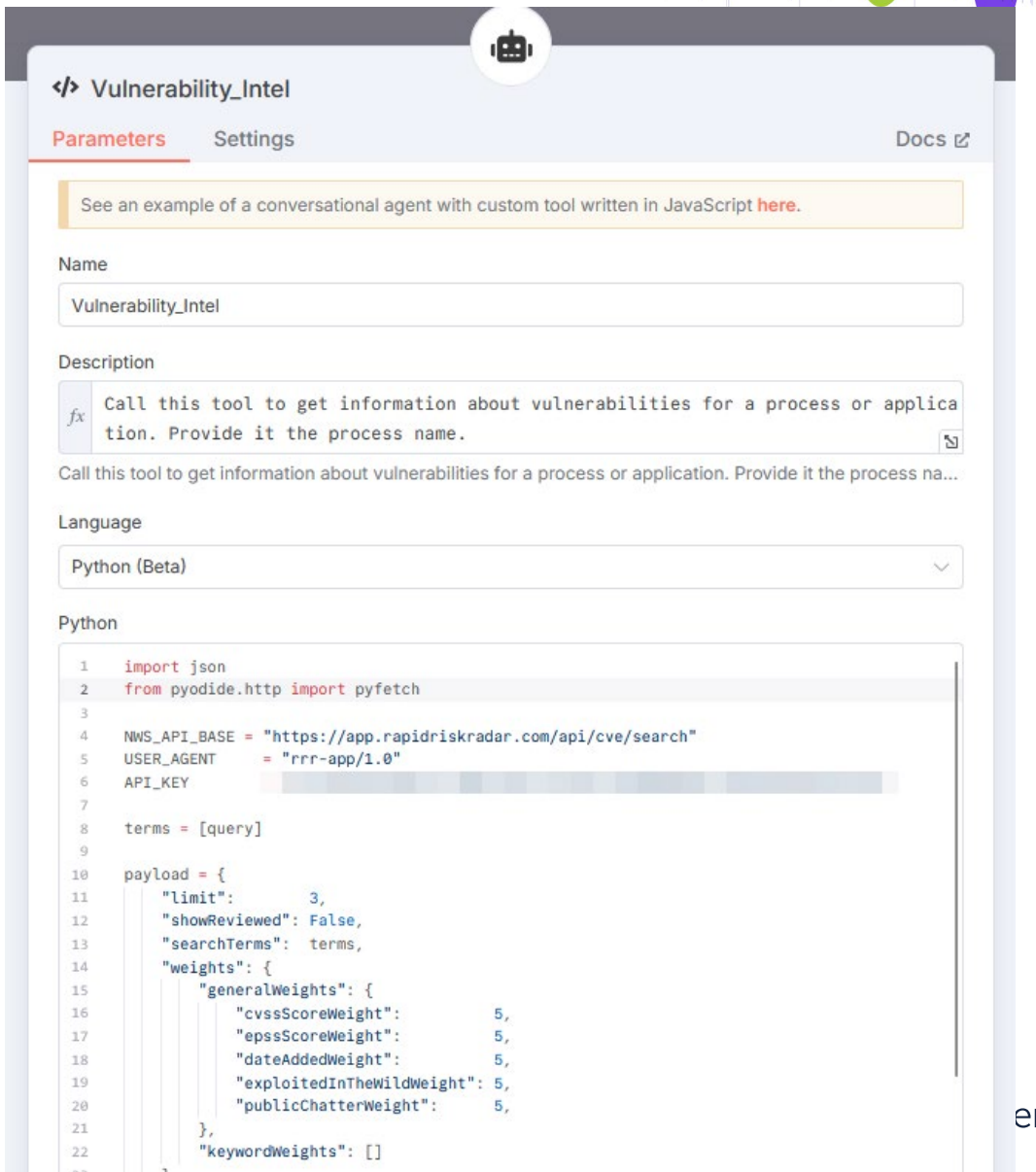
Waiting for message

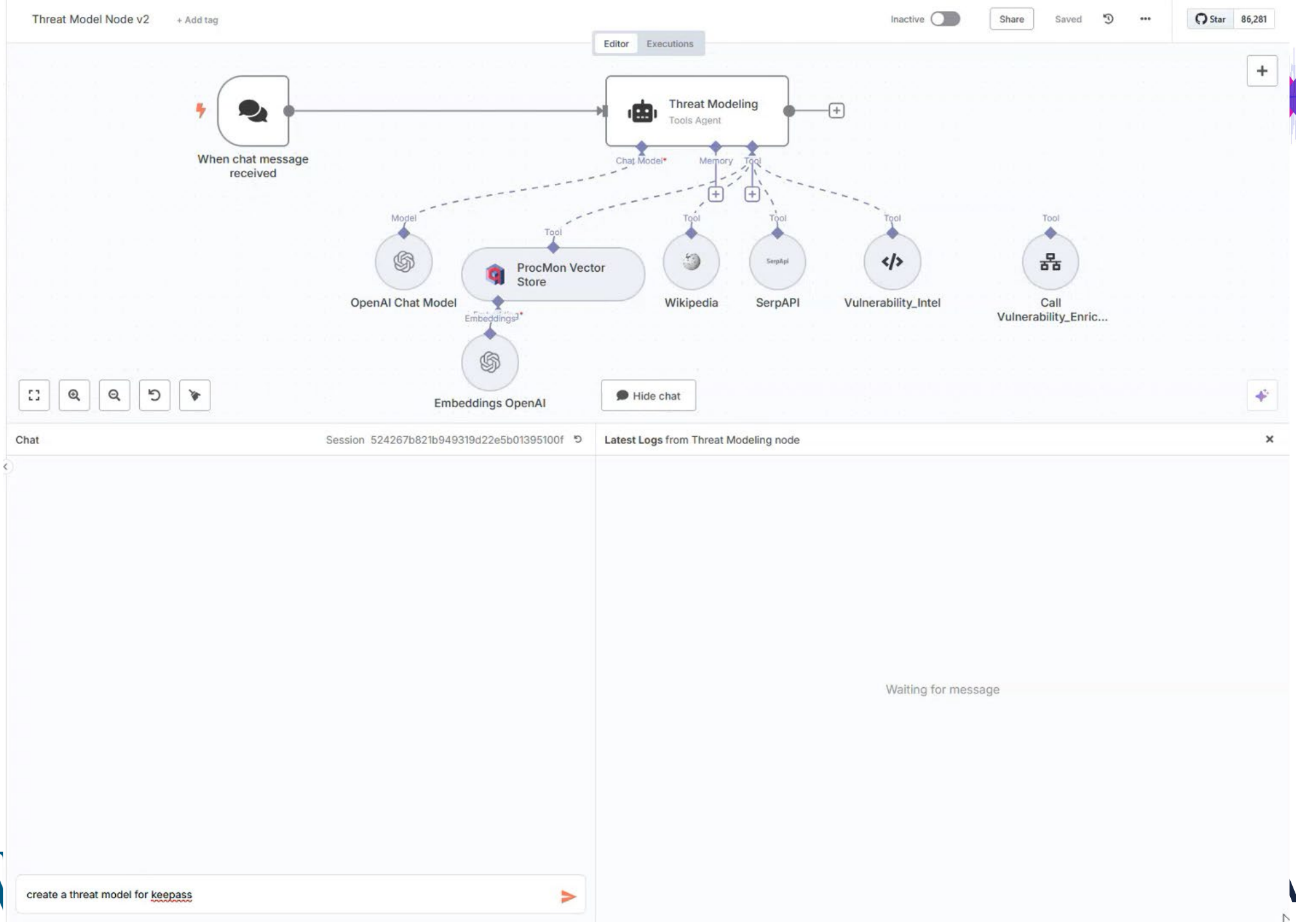
Create a threat model for keepass

MCP Inspired, Agent Optimized

https://github.com/rapidriskradar/RRR-MCP

```python
from typing import Any, Dict, Tuple, Optional
import httpx, sys
from mcp.server.fastmcp import FastMCP

# Initialize FastMCP server
mcp = FastMCP("rrr")

# Constants
NWS_API_BASE = "https://app.rapidriskradar.com/api/cve/search"
USER_AGENT = "rrr-app/1.0"
API_KEY = sys.argv[1]

async def make_request(data: Dict) -> dict[str, Any] | None:
    """Make a request to the RRR API with proper error handling."""
    headers = {
        "x-apikey": API_KEY,
        "Content-Type": "application/json",
        "User-Agent": USER_AGENT,
    }

    async with httpx.AsyncClient() as client:
        try:
            response = await client.post(NWS_API_BASE, headers=headers, json=data, timeout=30.
            0)
            response.raise_for_status()
            return response.json()
        except Exception:
            return None

@mcp.tool()
async def advanced_cve_search(cvss_range: Tuple[float, float], has_poc: Optional[bool],
exploitedinthewild: Optional[bool]) -> str:
    """Perform advanced searches.

    Args:
        cvss_range: (minimum, maximum) CVSS score to search for, default to [6, 10]
        has_poc: (optional) whether or not the CVEs have a proof of concept, leave as null to
        search both
        exploitedinthewild: (optional) whether or not the CVEs have been exploited in the
        wild, leave as null to search both
    """
```




Threat Model Node v2
+ Add tag
Inactive
Share
Saved
Star 86,281
Editor
Executions
When chat message received
Threat Modeling
Tools Agent
Chat Model*
Memory
Tool
Model
Tool
Tool
Tool
Tool
OpenAI Chat Model
ProcMon Vector Store
Wikipedia
SerpAPI
Vulnerability_Intel
Call Vulnerability_Enric...
Embeddings*
Embeddings OpenAI
Hide chat
Chat
Session 524267b821b949319d22e5b01395100f
Latest Logs from Threat Modeling node
Waiting for message
create a threat model for keepass

Output Analysis

EXECUTIVE SUMMARY

KeePass is a popular password management tool that has recently been identified with several critical vulnerabilities, including issues that allow unauthorized access to sensitive data stored within its databases. This threat model demonstrates the potential exploit vectors through process interactions and highlights the urgency for patching known vulnerabilities, particularly CVE-2023-32784, which has a CVSS score of 7.5. Immediate attention is recommended to secure the application and mitigate risks associated with its usage.

DATA SOURCE UTILIZATION

The analysis of KeePass involved data from process monitoring during its runtime and vulnerability intelligence. Significant findings include:

- **Process Interactions**: KeePass.exe exhibited multiple operations that interact with critical system resources, including file access for cryptographic libraries and registry settings essential for its functionality.
- **Vulnerabilities**: Observations linked to the following CVEs:
  - CVE-2023-32784: Allows recovery of cleartext passwords from memory.
  - CVE-2024-33901: Memory dump vulnerability leading to sensitive data exposure.
  - CVE-2023-24055: Risks from improper handling of configuration files.

Output Analysis

DREAD ASSESSMENT

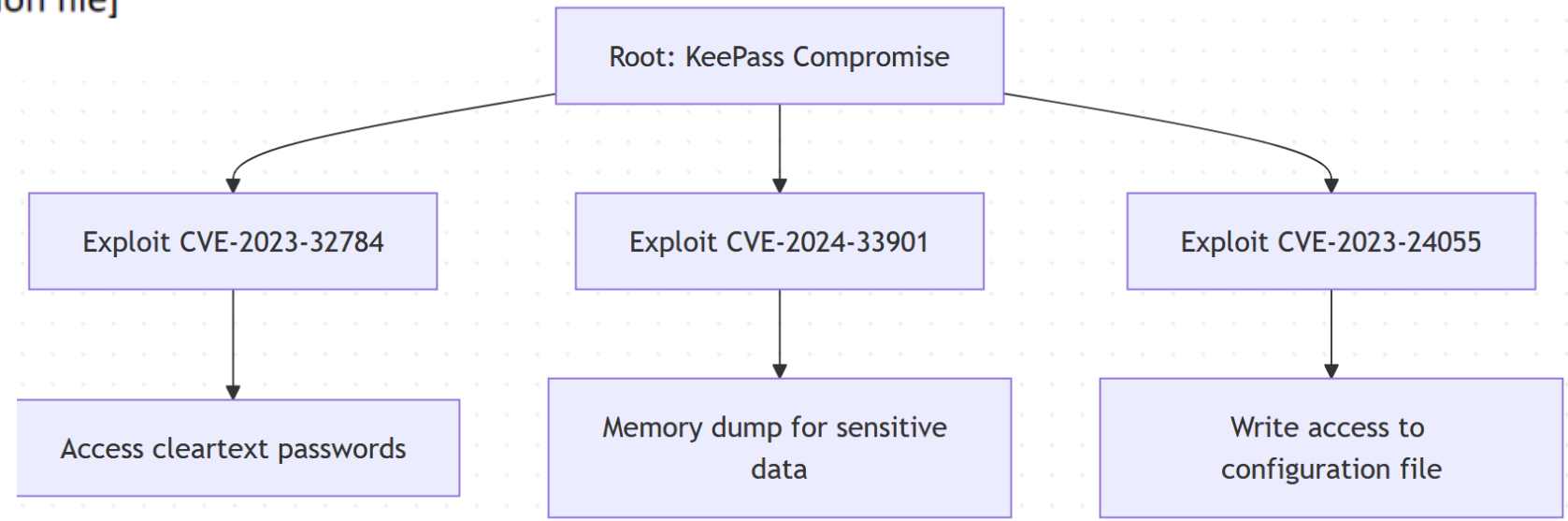
Component	Score (1-10)	Justification
Damage	8	Exposure of user passwords can lead to identity theft and access to sensitive information.
Reproducibility	7	Vulnerabilities can be reproduced in controlled environments, increasing likelihood of exploitation for both advanced users and malware.
Exploitability	7	Known exploits exist for some CVEs, and their descriptions indicate a moderate level of complexity for an attacker.
Affected users	9	KeePass is widely used globally, putting a large number of users at risk if vulnerabilities are exploited.
Discoverability	6	While some attacks can be detected, exploitation techniques may evade detection depending on the user's environment and tools.

Overall Risk Rating: (8 + 7 + 7 + 9 + 6) / 5 = **7.4**

Output Analysis

ATTACK TREE VISUALIZATION

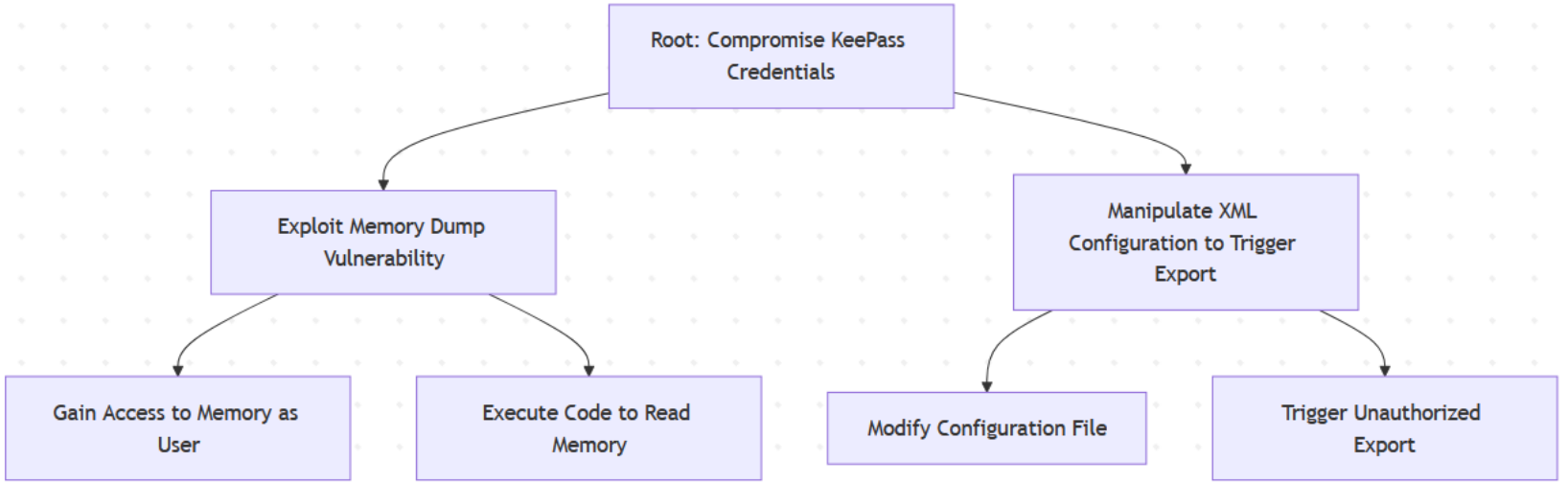
```
graph TD
    A[Root: KeePass Compromise] --> B[Exploit CVE-2023-32784]
    A --> C[Exploit CVE-2024-33901]
    A --> D[Exploit CVE-2023-24055]
    B --> E[Access cleartext passwords]
    C --> F[Memory dump for sensitive data]
    D --> G[Write access to configuration file]
```



Output Analysis

ATTACK TREE VISUALIZATION

```
graph TD
    A[Root: Compromise KeePass Credentials] --> B[Exploit Memory Dump Vulnerability]
    A --> C[Manipulate XML Configuration to Trigger Export]
    B --> D1[Gain Access to Memory as User]
    B --> D2[Execute Code to Read Memory]
    C --> E1[Modify Configuration File]
    C --> E2[Trigger Unauthorized Export]
```


Root: Compromise KeePass Credentials
Exploit Memory Dump Vulnerability
Manipulate XML Configuration to Trigger Export
Gain Access to Memory as User
Execute Code to Read Memory
Modify Configuration File
Trigger Unauthorized Export

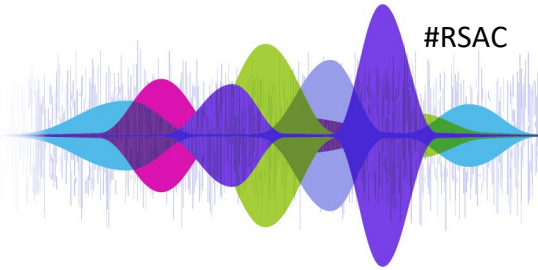
Output Analysis

MITIGATION ROADMAP

1. **Immediate Actions**:
   - Upgrade KeePass to the latest version to address CVE-2023-32784 and related vulnerabilities.
   - Apply system patches and updates that address underlying cryptographic vulnerabilities associated with the library used.
2. **Long-Term Strategies**:
   - Implement regular vulnerability scanning and monitoring of KeePass installations.
   - Educate users about securing password management practices, including the importance of keeping the application updated.
3. **Detection Strategies**:
   - Monitor for abnormal registry access patterns and file access to sensitive DLLs related to KeePass operations.
   - Set up alerts for excessive memory usage from KeePass processes that may indicate exploitation attempts.

References to specific patches, updates, and further resources will be needed to ensure that the implementation of these mitigation strategies is effective.

AI Strengths vs. Blind Spots



Strengths

- Shifting through lots of data
- Summarizing
- Using tools
- Formatting
- Pattern Recognition
- Speed and Scale

Blind Spots

- Ambiguity Handling
- Complexity
- Deterministic
- True Creativity
- Unstructured Exploration
- Common Sense Reasoning

Apply What You Have Learned Today

https://github.com/aboutsecurity/rsac2025

- Next week you should:
  - Review the SANS Product Security Testing methodology
  - Identify and prioritize critical assets
  - Threat model based on high value assets
- In the first three months following this presentation, you should:
  - Develop a plan to implement product security testing on critical assets
  - Define a plan to integrate AI and automation into product security, while keeping the expert in the loop
- Within six months, you should:
  - Define a plan to integrate product security testing on new assets
  - Executing product security tests on critical assets
  - Updating your company threat models and implementing product-specific mitigations

RSAC™ 2025 Conference

Many Voices.
One Community.

Thank you!